UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/808,185	03/24/2004	John C. Lallier	34826-1017	4569

41881 7590 01/02/2009

BRANDON N. SKLAR. ESQ. (PATENT PROSECUTION)
KAYE SCHOLER, LLP
425 PARK AVENUE
NEW YORK, NY 10022-3598

EXAMINER	
DWIVEDI, MAHESH H	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
01/02/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/808,185	LALLIER, JOHN C.	
	Examiner	Art Unit	
	MAHESH H. DWIVEDI	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _15 December 2008_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _19-26,52-59,79 and 88-90_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _19-26,52-59,79 and 88-90_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _14 June 2004_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 12/15/2008 has been entered.

Remarks

2. Receipt of the Applicant's amendments filed on 12/15/2008 is acknowledged. The amendment includes the amending of claims 19 and 89-90, and the cancellation of claims 1-18, 27-51, 60-78, 80-87, and 91.

Specification

3. The objections raised in the Office Action mailed on 09/15/2008 have been overcome by Applicant's Amendments received on 12/15/2008.

Claim Rejections - 35 USC § 112

4. The rejections raised in the Office Action mailed on 09/15/2008 have been overcome by Applicant's Amendments received on 12/15/2008.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 19-21, 23-25, 52-54, 56-58, 79, 88, 90, 92-96, and 98-100 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eshel et al.** (U.S. PGPUB 2003/0158862) and in view of **Hubis et al.** (U.S. Patent 6,182,198).

7. Regarding claim 19, **Eshel** teaches a method comprising:

A) storing in a target storage device a plurality of target data files corresponding respectively to respective ones of a plurality of source data files stored in a source storage device (Paragraph 130);

B) storing in each respective target data file information identifying the corresponding source data file identifying the corresponding source data file (Paragraph 132);

C) activating a de-migration procedure to copy data from the source storage device to the target storage device, after target data files have been stored for all source data files in the plurality (Paragraph 130);

E) examining, in a target data file corresponding to the specified data file, selected information identifying a corresponding source data file (Paragraph 132);

F) retrieving the requested data from the corresponding source data file (Paragraph 127); and

G) providing the requested data to the host device (Paragraph 127).

The examiner notes that **Eshel** teaches **"storing in a target storage device a plurality of target data files corresponding respectively to respective ones of a plurality of source data files stored in a source storage device"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)." (Paragraph 130). The examiner further notes that **Eshel** teaches **"storing in each respective target data file information identifying the corresponding source data file identifying the corresponding source data file"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the

different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the last applied or restored snapshot" (Paragraph 132). The examiner further notes that **Eshel** teaches **"activating a de-migration procedure to copy data from the source storage device to the target storage device, after target data files have been stored for all source data files in the plurality"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more

recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130). The examiner further notes that **Eshel** teaches **"examining, in a target data file corresponding to the specified data file, selected information identifying a corresponding source data file"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied

snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the last applied or restored snapshot" (Paragraph 132). The examiner further notes that **Eshel** teaches **"retrieving the requested data from the corresponding source data file"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"providing the requested data to the host device"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127).

Eshel does not explicitly teach:

D) receiving from a host device, <u>at</u> the target storage device, a request specifying a data file, while the de-migration procedure is executing.

Hubis, however, teaches **"receiving from a host device, <u>at</u> the target storage device, a request specifying a data file, while the de-migration procedure is executing"** as "The method executes a read operation during the snapshot backup by processing the read operation submitted to the backup logic unit by accessing the requested data of the read operation from the log system drive if the requested data is available from the log system drive and returning the requested data to a requester, if not, accessing the requested data from the primary system drive and returning the requested data to the requester" (Column 2, lines 31-38), and "In backup operation, shown in FIG. 6, the backup LUN 213 is now available, accepts only read operations, and can access data from both primary system drive 210 and log system drive 212. In response to a read request 217 on backup LUN 213, the controller (not shown) first checks to see if the data to fulfill the request is available on log system drive 212. If so, the information is read 221 from log system drive 212 and returned to the requester (not shown). If not, the data is read 220 from primary system drive 210 and returned to the requestor. Any write operations to backup LUN 213 are rejected"

(Column 5, lines 21-32), and “The copy process 236 can be monitored by a System Drive Copy Status command. While the copy is being performed, the backup LUNs 234 associated with backup system drive 231 responds with busy status. Any LUNs (not shown) associated with log system drive 233 operates as a normal snapshot backup LUN, with read-only capability. When the copy is complete, backup system drive 231 responds normally to all read and write commands, and the snapshot backup process terminates” (Column 7, lines 16-24)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Hubis’s** would have allowed **Eshel’s** to provide a new and improved method and apparatus for generating a snapshot backup that can allow read and write operations to occur while a snapshot backup is in progres, as noted by **Hubis** (Column 2, lines 5-8).

Regarding claim 20, **Eshel** further teaches a method comprising:
A) wherein the source data file is stored in a file volume on the source storage device (Paragraph 129, Figure 15a).

The examiner notes that **Eshel** teaches **“wherein the source data file is stored in a file volume on the source storage device”** as “A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File

system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot dataset copy 1510 and a second snapshot dataset copy 1512 to support standby data storage operations" (Paragraph 129).

Regarding claim 21, **Eshel** further teaches a method comprising:
A) wherein the target data file is stored in a file volume on the target storage device (Paragraph 129, Figure 15a).

The examiner notes that **Eshel** teaches **"wherein the target data file is stored in a file volume on the target storage device"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot dataset copy 1510 and a second snapshot dataset copy 1512 to support standby data storage operations" (Paragraph 129).

Regarding claim 23, **Eshel** further teaches a method comprising:

A) wherein the target storage device comprises a file server (Paragraph 49).

The examiner notes that **Eshel** teaches **"wherein the target data file is stored in a file volume on the target storage device"** as "In another embodiment of the present invention, the computer systems of file system 102 and backup 108 are a server such as one or more computers executing operating systems such as SunOS or AIX, such as SUN Ultra workstations running the SunOS operating system or IBM RS/6000 workstations and servers running the AIX operating system" (Paragraph 49).

Regarding claim 24, **Eshel** further teaches a method comprising:

A) wherein the request is received from the host device via a network (Paragraph 129).

The examiner notes that **Eshel** teaches **"wherein the request is received from the host device via a network"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems" (Paragraph 129).

Regarding claim 25, **Eshel** further teaches a method comprising:

A) wherein the selected information in a respective target data file identifies a logical location of the corresponding source data file in a source file volume (Paragraph 132).

The examiner notes that **Eshel** teaches **"wherein the selected information in a respective target data file identifies a logical location of the corresponding source data file in a source file volume"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The

snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the last applied or restored snapshot" (Paragraph 132).

Regarding claim 52, **Eshel** teaches a system comprising:

A) a target storage device configured to store data files (Paragraph 130);

B) a source storage device configured to store data files (Paragraph 130);

C) at least one processor configured to: store in the target storage device a plurality of target data files corresponding respectively to respective ones of a plurality of source data files in the source storage device (Paragraph 130);
D) store in each respective target data file information identifying the corresponding source data file (Paragraph 132);
E) activate a de-migration procedure to copy data from the source storage device to the target storage device, after the target data files have been stored for all source data files in the plurality (Paragraph 130); and
F) wherein the target storage device is further configured to: receive from a host device a request specifying a data file, while the de-migration procedure is executing (Paragraph 127);
G) examine, a target data file corresponding to the specified data file, selected information identifying a corresponding source data file (Paragraph 132);
H) wherein the at least one processor is further configured to: retrieve requested data from the corresponding source data file (Paragraph 127); and
I) provide the requested data to the host device (Paragraph 127).

The examiner notes that **Eshel** teaches **"a target storage device configured to store data files"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)." The examiner further notes that **Eshel** teaches **"a source storage device configured to store data files"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)." The examiner further notes that **Eshel** teaches **"at least one processor configured to: store in the target storage device a plurality of target data files corresponding respectively to respective ones of a plurality of source data files in the**

source storage device" as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)." The examiner further notes that **Eshel** teaches **"store in each respective target data file information identifying the corresponding source data file"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of

changes that corresponds to the changes made since the last applied or restored snapshot" (Paragraph 132). The examiner notes that **Eshel** teaches **"activate a de-migration procedure to copy data from the source storage device to the target storage device, after the target data files have been stored for all source data files in the plurality"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130). The examiner further notes that **Eshel** teaches **"wherein the target storage device is further configured to: receive from a host device a request specifying a data file, while the de-migration procedure is executing"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"examine, a target data file corresponding to the specified data file, selected information identifying a corresponding source data file"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein

to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the last applied or restored snapshot" (Paragraph 132). The examiner further notes that **Eshel** teaches **"wherein the at least one processor is further configured to: retrieve requested data from the corresponding source data file"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"providing the requested data to the host device"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127).

Regarding claim 53, **Eshel** further teaches a system comprising:

A) wherein the source data file is stored in a file volume on the source storage device (Paragraph 129, Figure 15a).

The examiner notes that **Eshel** teaches **"wherein the source data file is stored in a file volume on the source storage device"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot dataset copy 1510 and a second snapshot dataset copy 1512 to support standby data storage operations" (Paragraph 129).

Regarding claim 54, **Eshel** further teaches a system comprising:

A) wherein the target data file is stored in a file volume on the target storage device (Paragraph 129, Figure 15a).

The examiner notes that **Eshel** teaches **"wherein the target data file is stored in a file volume on the target storage device"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and

a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot dataset copy 1510 and a second snapshot dataset copy 1512 to support standby data storage operations" (Paragraph 129).

Regarding claim 56, **Eshel** further teaches a system comprising:
A) wherein the target storage device comprises a file server (Paragraph 49).

The examiner notes that **Eshel** teaches **"wherein the target data file is stored in a file volume on the target storage device"** as "In another embodiment of the present invention, the computer systems of file system 102 and backup 108 are a server such as one or more computers executing operating systems such as SunOS or AIX, such as SUN Ultra workstations running the SunOS operating system or IBM RS/6000 workstations and servers running the AIX operating system" (Paragraph 49).

Regarding claim 57, **Eshel** further teaches a system comprising:
A) wherein the request is received from the host device via a network (Paragraph 129).

The examiner notes that **Eshel** teaches **"wherein the request is received from the host device via a network"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A.

This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems" (Paragraph 129).

Regarding claim 58, **Eshel** further teaches a system comprising:

A) wherein the selected information in a respective target data file identifies a logical location of the corresponding source data file in a source file volume (Paragraph 132).

The examiner notes that **Eshel** teaches **"wherein the selected information in a respective target data file identifies a logical location of the corresponding source data file in a source file volume"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file

system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the last applied or restored snapshot" (Paragraph 132).

Regarding claim 79, **Eshel** further teaches a method comprising:
A) copying the identified source data file from the source storage device to the target storage device (Paragraph 130).

The examiner notes that **Eshel** teaches **"copying the identified source data file from the source storage device to the target storage device"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130).

Regarding claim 88, **Eshel** further teaches a method comprising:
A) activating a de-migration procedure to copy source data files form the source storage device to locations of corresponding target data files (Paragraph 49).

The examiner notes that **Eshel** teaches **"activating a de-migration procedure to copy source data files form the source storage device to locations of corresponding target data files"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130).

Regarding claim 90, **Eshel** teaches a method comprising:

A) storing in a target storage device a plurality of target data files corresponding respectively to respective ones of a plurality of source data files stored in a source storage device (Paragraphs 130 and 132);

B) storing in each respective target data file information identifying the corresponding source data file (Paragraph 132);

C) activating a de-migration procedure to copy source data files from the source storage device to locations of the corresponding target data files in the target storage device (Paragraph 130);

E) copying selected data from a source data file identified within the specified target data file to the specified target storage device, in response to the data processing request (Paragraphs 127 and 130).

The examiner notes that **Eshel** teaches **"storing in a target storage device a plurality of target data files corresponding respectively to**

respective ones of a plurality of source data files stored in a source storage device" as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)." (Paragraph 130) and "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the

last applied or restored snapshot" (Paragraph 132). The examiner further notes that **Eshel** teaches **"storing in each respective target data file information identifying the corresponding source data file"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the last applied or restored snapshot" (Paragraph 132). The examiner notes that **Eshel** teaches **"activating a de-migration procedure to copy source data files from the source storage**

device to locations of the corresponding target data files in the target storage device" as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130). The examiner further notes that **Eshel** teaches **"copying selected data from a source data file identified within the specified target data file to the specified target storage device, in response to the data processing request"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127) and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the

standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130).

Eshel does not explicitly teach:

D) receiving, at the target storage device, a data processing request specifying a target data file while the de-migration procedure is executing.

Hubis, however, teaches **"receiving, at the target storage device, a data processing request specifying a target data file while the de-migration procedure is executing"** as "The method executes a read operation during the snapshot backup by processing the read operation submitted to the backup logic unit by accessing the requested data of the read operation from the log system drive if the requested data is available from the log system drive and returning the requested data to a requester, if not, accessing the requested data from the primary system drive and returning the requested data to the requester" (Column 2, lines 31-38), and "In backup operation, shown in FIG. 6, the backup LUN 213 is now available, accepts only read operations, and can access data from both primary system drive 210 and log system drive 212. In response to a read request 217 on backup LUN 213, the controller (not shown) first checks to see if the data to fulfill the request is available on log system drive 212. If so, the information is read 221 from log system drive 212 and returned to the requester (not shown). If not, the data is read 220 from primary system drive 210 and returned to the requestor. Any write operations to backup LUN 213 are rejected" (Column 5, lines 21-32), and "The copy process 236 can be monitored by a System Drive Copy Status command. While the copy is being performed, the backup LUNs 234 associated with backup system drive 231 responds with busy status. Any LUNs (not shown) associated with log system drive 233 operates as a normal snapshot backup LUN, with read-only capability. When the copy is complete, backup system drive 231 responds normally to all read and write commands, and the snapshot backup process terminates" (Column 7, lines 16-24)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Hubis's** would have allowed **Eshel's** to provide a new and improved method and apparatus for generating a snapshot backup that can allow read and write operations to occur while a snapshot backup is in progres, as noted by **Hubis** (Column 2, lines 5-8).

Regarding claim 92, **Eshel** further teaches a method comprising:
A) prior to activating the de-migration procedure: receiving by a processor, from the host device, at least one first data processing request (Paragraph 47); and
B) sending the at least one first data processing request to the source storage device (Paragraph 47); and
C) after activating the de-migration procedure: receiving by the processor, from the host side device, at least one second data processing request (Paragraph 127); and
D) sending the at least one second data processing request to the target storage device (Paragraphs 127 and 130).

The examiner further notes that **Eshel** teaches **"receiving, by the target storage device, a data processing request specifying a target data file while the de-migration procedure is executing"** as "Referring now in more detail to the drawings in which like numerals refer to like parts throughout several views, an exemplary overall system architecture 100 in which exemplary embodiments of the present invention operate is illustrated in FIG. 1. The exemplary embodiments of the present invention operate within or in conjunction with a file system 102 that is used to store one or more data files. The exemplary embodiments of the present invention capture and maintain one or more snapshot datasets 104, which are described in detail below. The computer, or client information processing system, upon which the file system 102 exists in this exemplary overall system architecture 100 is connected to other computers and data processing systems via network 106. One application for the exemplary

embodiments of the present invention is to support efficient processing for backing-up data contained on a data storage system. An exemplary backup system 108 is shown in the exemplary overall system architecture 100. The exemplary backup system 108 is used to maintain a backup, which is a copy of all of the data contained within the file system 102. One use of the snapshot 104 is to efficiently communicate and store backup datasets upon remote backup systems, such as backup system 108. The snapshot data captured and maintained by the exemplary embodiments of the present invention are used for a large variety of uses beyond performing data backups. The snapshot data is used, for example, to recover accidentally deleted files or to retrieve data that has been overwritten either accidentally or intentionally" (Paragraph 47). The examiner further notes that **Eshel** teaches **"copying selected data from a source data file identified within the specified target data file to the specified target storage device, in response to the data processing request"** as "Referring now in more detail to the drawings in which like numerals refer to like parts throughout several views, an exemplary overall system architecture 100 in which exemplary embodiments of the present invention operate is illustrated in FIG. 1. The exemplary embodiments of the present invention operate within or in conjunction with a file system 102 that is used to store one or more data files. The exemplary embodiments of the present invention capture and maintain one or more snapshot datasets 104, which are described in detail below. The computer, or client information processing system, upon which the file system 102 exists in this exemplary overall system architecture 100 is connected to other computers and data processing systems via network 106. One application for the exemplary embodiments of the present invention is to support efficient processing for backing-up data contained on a data storage system. An exemplary backup system 108 is shown in the exemplary overall system architecture 100. The exemplary backup system 108 is used to maintain a backup, which is a copy of all of the data contained within the file system 102. One use of the snapshot 104 is to efficiently communicate and

store backup datasets upon remote backup systems, such as backup system 108. The snapshot data captured and maintained by the exemplary embodiments of the present invention are used for a large variety of uses beyond performing data backups. The snapshot data is used, for example, to recover accidentally deleted files or to retrieve data that has been overwritten either accidentally or intentionally" (Paragraph 47). The examiner further notes that **Eshel** teaches **"after activating the de-migration procedure: receiving by the processor, from the host side device, at least one second data processing request"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"sending the at least one second data processing request to the target storage device"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127) and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130).

Regarding claim 93, **Eshel** further teaches a method comprising:

A) storing in each respective target data file a respective pointer identifying the corresponding source data file (Paragraphs 131-133).

The examiner notes that **Eshel** teaches **"storing in each respective target data file a respective pointer identifying the corresponding source data file"** as "The exemplary embodiments of the present invention use snapshot tags to identify each snapshot and the file system from which that snapshot was captured. The snapshot tag notation used herein consists of the format (A:S1) wherein the first element, "A" in this example, identifies the file system and the second element, "S1" in this example, is the snapshot identifier for that snapshot. This allows the different file systems in the hot standby system described herein to capture snapshots at different times and only use a subset of those snapshots to synchronize the data between those file systems. The file systems of the exemplary embodiments generate a set of changes between snapshots that are captured for that file system. These sets of changes include a pair of tags to identify the snapshots between which the changes were determined. As an example, a snapshot tag pair (A:S2, A:S3) is included within a set of changes that were generated as the changes that occurred between snapshot S2 and snapshot S3 that were captured on file system A. This set of changes is only able to be successfully applied to a file system that has been restored to the state of snapshot S2 from file system A. For example, if file system B receives this snapshot and snapshot S2 from file system A has not been restored to file system B or changes have not been applied to file system B that resulted in file system B having the state of snapshot (A:S2), application of the set of changes with the snapshot tag pair (A:S2,A:S3) is inappropriate. A file system discards a set of changes that is received and does not have a snapshot pair that starts with a snapshot tag that corresponds to the most recently restored or applied snapshot to that file system. Exemplary systems identify the last applied or restored snapshot and request from the other file system the set of changes that corresponds to the changes made since the last applied or restored snapshot. The snapshot tags are stored in the snapshot and also in each of the

file systems. The snapshot tags stored in the file systems are stored in the superblock for the file system and identify the latest snapshot that was restored in order to establish a base file system and the snapshot tag of the latest snapshot that has been applied to the base file system is also stored in the superblock of the file system. The snapshot tag in the file system is compared to the snapshot tag of a newly received snapshot or set of changes before that new snapshot or set of changes is applied to the file system" (Paragraphs 132-133).

Regarding claim 94, **Eshel** further teaches a method comprising:

A) examining a selected pointer stored in the target data file (Paragraphs 131 and 133); and

B) copying the corresponding source data file from the source storage device to the target storage device, based at least on information in the selected pointer (Paragraphs 131 and 133).

The examiner notes that **Eshel** teaches **"examining a selected pointer stored in the target data file"** as "Maintenance of the standby file system is facilitated in the exemplary embodiments by maintaining snapshot tags that uniquely identify both the different snapshots that recorded the state of each of the file systems at different times and that identify the set of changes that are generated between two snapshots. The snapshot tags are used to coordinate proper data synchronization between the mirror file system and the active file system when switching the mirror file system from a read only file system to the active read/write file system by ensuring that the latest snapshot is applied after a failure disables the original file system. Once the initial mirror file system becomes the active file system that is used by client processors (i.e., the "new original" file system), snapshots are captured of the new original file system and snapshot tags are used to restore the previous original file system, which is now the mirror, to maintain the original file system as the new standby, or mirror, file system" (Paragraph 131) and "The snapshot tags are stored in the snapshot and also in each of the file systems. The snapshot tags stored in the file systems are

stored in the superblock for the file system and identify the latest snapshot that was restored in order to establish a base file system and the snapshot tag of the latest snapshot that has been applied to the base file system is also stored in the superblock of the file system. The snapshot tag in the file system is compared to the snapshot tag of a newly received snapshot or set of changes before that new snapshot or set of changes is applied to the file system. Only a snapshot or a set of changes with a base snapshot tag that corresponds to the base snapshot that has most recently been used on the file system is applied to the file system. Once a snapshot from a source file system is applied to a mirror file system, another snapshot is captured of the mirror file system that puts it in sync with the original file system. The file systems of the exemplary embodiments store the snapshot tags for the last restored or applied data in the superblock of the file system. The snapshot tags identify the source file system and the snapshot identifier of the last snapshot on the remote system that was copied to this file system. An example use of this data is in the event that a series of snapshot updates are lost or corrupted when received by a file system. In the event that a file system does not properly receive one or more sets of changes, the last properly applied set of changes is determined and the remote file system is queried for the set of changes that were made to that file system since the snapshot that corresponds to the last set of data that was properly restored or applied" (Paragraph 133). The examiner further notes that **Eshel** teaches **"copying the corresponding source data file from the source storage device to the target storage device, based at least on information in the selected pointer"** as "Maintenance of the standby file system is facilitated in the exemplary embodiments by maintaining snapshot tags that uniquely identify both the different snapshots that recorded the state of each of the file systems at different times and that identify the set of changes that are generated between two snapshots. The snapshot tags are used to coordinate proper data synchronization between the mirror file system and the active file system when switching the mirror file system from a read only file system to the active

read/write file system by ensuring that the latest snapshot is applied after a failure disables the original file system. Once the initial mirror file system becomes the active file system that is used by client processors (i.e., the "new original" file system), snapshots are captured of the new original file system and snapshot tags are used to restore the previous original file system, which is now the mirror, to maintain the original file system as the new standby, or mirror, file system" (Paragraph 131) and "The snapshot tags are stored in the snapshot and also in each of the file systems. The snapshot tags stored in the file systems are stored in the superblock for the file system and identify the latest snapshot that was restored in order to establish a base file system and the snapshot tag of the latest snapshot that has been applied to the base file system is also stored in the superblock of the file system. The snapshot tag in the file system is compared to the snapshot tag of a newly received snapshot or set of changes before that new snapshot or set of changes is applied to the file system. Only a snapshot or a set of changes with a base snapshot tag that corresponds to the base snapshot that has most recently been used on the file system is applied to the file system. Once a snapshot from a source file system is applied to a mirror file system, another snapshot is captured of the mirror file system that puts it in sync with the original file system. The file systems of the exemplary embodiments store the snapshot tags for the last restored or applied data in the superblock of the file system. The snapshot tags identify the source file system and the snapshot identifier of the last snapshot on the remote system that was copied to this file system. An example use of this data is in the event that a series of snapshot updates are lost or corrupted when received by a file system. In the event that a file system does not properly receive one or more sets of changes, the last properly applied set of changes is determined and the remote file system is queried for the set of changes that were made to that file system since the snapshot that corresponds to the last set of data that was properly restored or applied" (Paragraph 133).

Regarding claim 95, **Eshel** further teaches a method comprising:

A) replacing the target data file with the copied source data file (Paragraphs 133 and 143).

The examiner notes that **Eshel** teaches **"replacing the target data file with the copied source data file"** as "The snapshot tags are stored in the snapshot and also in each of the file systems. The snapshot tags stored in the file systems are stored in the superblock for the file system and identify the latest snapshot that was restored in order to establish a base file system and the snapshot tag of the latest snapshot that has been applied to the base file system is also stored in the superblock of the file system. The snapshot tag in the file system is compared to the snapshot tag of a newly received snapshot or set of changes before that new snapshot or set of changes is applied to the file system. Only a snapshot or a set of changes with a base snapshot tag that corresponds to the base snapshot that has most recently been used on the file system is applied to the file system. Once a snapshot from a source file system is applied to a mirror file system, another snapshot is captured of the mirror file system that puts it in sync with the original file system. The file systems of the exemplary embodiments store the snapshot tags for the last restored or applied data in the superblock of the file system. The snapshot tags identify the source file system and the snapshot identifier of the last snapshot on the remote system that was copied to this file system. An example use of this data is in the event that a series of snapshot updates are lost or corrupted when received by a file system. In the event that a file system does not properly receive one or more sets of changes, the last properly applied set of changes is determined and the remote file system is queried for the set of changes that were made to that file system since the snapshot that corresponds to the last set of data that was properly restored or applied" (Paragraph 133) and "After file system A is initialized and becomes the standby file system, file system B then generates, at step 1560, a set of changes between the last snapshot that was received from file system A, snapshot 1 in this example, and communicates that set of changes to file system A. This set of

changes contains the snapshot tag pair (A:S1, B:S3). File system A receives, at step 1562, this generated set of changes from file system B and applies those changes to the data stored on file system A in order to establish a copy of the data of file system B. After applying these changes, file system A then captures a snapshot, snapshot 3 in this example, of the data on that file system. If a previous snapshot of file system A in this example does not exist on file system a, then an entire backup dataset of file system B is generated at file system B, communicated to file system A and restored on file system A" (Paragraph 143).

Regarding claim 96, **Eshel** further teaches a method comprising:

A) examining a selected pointer stored in the target data file (Paragraphs 131 and 133);

B) determining a size of the corresponding source data file (Paragraphs 54 and 143); and

C) copying the corresponding source data file from the source storage device to the target storage device, based at least on information in the selected pointer and on the size of the corresponding source data file (Paragraphs 54, 131, and 143).

The examiner notes that **Eshel** teaches **"examining a selected pointer stored in the target data file"** as "Maintenance of the standby file system is facilitated in the exemplary embodiments by maintaining snapshot tags that uniquely identify both the different snapshots that recorded the state of each of the file systems at different times and that identify the set of changes that are generated between two snapshots. The snapshot tags are used to coordinate proper data synchronization between the mirror file system and the active file system when switching the mirror file system from a read only file system to the active read/write file system by ensuring that the latest snapshot is applied after a failure disables the original file system. Once the initial mirror file system becomes the active file system that is used by client processors (i.e., the "new original" file system), snapshots are captured of the new original file system and

snapshot tags are used to restore the previous original file system, which is now the mirror, to maintain the original file system as the new standby, or mirror, file system" (Paragraph 131) and "The snapshot tags are stored in the snapshot and also in each of the file systems. The snapshot tags stored in the file systems are stored in the superblock for the file system and identify the latest snapshot that was restored in order to establish a base file system and the snapshot tag of the latest snapshot that has been applied to the base file system is also stored in the superblock of the file system. The snapshot tag in the file system is compared to the snapshot tag of a newly received snapshot or set of changes before that new snapshot or set of changes is applied to the file system. Only a snapshot or a set of changes with a base snapshot tag that corresponds to the base snapshot that has most recently been used on the file system is applied to the file system. Once a snapshot from a source file system is applied to a mirror file system, another snapshot is captured of the mirror file system that puts it in sync with the original file system. The file systems of the exemplary embodiments store the snapshot tags for the last restored or applied data in the superblock of the file system. The snapshot tags identify the source file system and the snapshot identifier of the last snapshot on the remote system that was copied to this file system. An example use of this data is in the event that a series of snapshot updates are lost or corrupted when received by a file system. In the event that a file system does not properly receive one or more sets of changes, the last properly applied set of changes is determined and the remote file system is queried for the set of changes that were made to that file system since the snapshot that corresponds to the last set of data that was properly restored or applied" (Paragraph 133). The examiner further notes that **Eshel** teaches **"determining a size of the corresponding source data file"** as "Inodes: metadata elements that contain file attributes (e.g., owner, access permissions, modified time, file size), and also specify the physical disk addresses of data blocks (for small files) or indirect blocks (for large files with more data blocks than the number of disk addresses that fit in an inode). In the description of the

exemplary embodiments of present invention, the collection of inodes is referred to as an "node file." The exemplary embodiments store inode files as a regular file (inode plus indirect blocks), but other embodiments use different representations of the collection of inodes. The collection of some or all of the information contained within the inode is referred to as "node information (Paragraph 54) and “After file system A is initialized and becomes the standby file system, file system B then generates, at step 1560, a set of changes between the last snapshot that was received from file system A, snapshot 1 in this example, and communicates that set of changes to file system A. This set of changes contains the snapshot tag pair (A:S1, B:S3). File system A receives, at step 1562, this generated set of changes from file system B and applies those changes to the data stored on file system A in order to establish a copy of the data of file system B. After applying these changes, file system A then captures a snapshot, snapshot 3 in this example, of the data on that file system. If a previous snapshot of file system A in this example does not exist on file system a, then an entire backup dataset of file system B is generated at file system B, communicated to file system A and restored on file system A” (Paragraph 143). The examiner further notes that **Eshel** teaches **“determining a size of the corresponding source data file”** as “Inodes: metadata elements that contain file attributes (e.g., owner, access permissions, modified time, file size), and also specify the physical disk addresses of data blocks (for small files) or indirect blocks (for large files with more data blocks than the number of disk addresses that fit in an inode). In the description of the exemplary embodiments of present invention, the collection of inodes is referred to as an "node file." The exemplary embodiments store inode files as a regular file (inode plus indirect blocks), but other embodiments use different representations of the collection of inodes. The collection of some or all of the information contained within the inode is referred to as "node information (Paragraph 54), “Maintenance of the standby file system is facilitated in the exemplary embodiments by maintaining snapshot tags that uniquely identify both the different snapshots that recorded the state of each of

the file systems at different times and that identify the set of changes that are generated between two snapshots. The snapshot tags are used to coordinate proper data synchronization between the mirror file system and the active file system when switching the mirror file system from a read only file system to the active read/write file system by ensuring that the latest snapshot is applied after a failure disables the original file system. Once the initial mirror file system becomes the active file system that is used by client processors (i.e., the "new original" file system), snapshots are captured of the new original file system and snapshot tags are used to restore the previous original file system, which is now the mirror, to maintain the original file system as the new standby, or mirror, file system" (Paragraph 131), and "After file system A is initialized and becomes the standby file system, file system B then generates, at step 1560, a set of changes between the last snapshot that was received from file system A, snapshot 1 in this example, and communicates that set of changes to file system A. This set of changes contains the snapshot tag pair (A:S1, B:S3). File system A receives, at step 1562, this generated set of changes from file system B and applies those changes to the data stored on file system A in order to establish a copy of the data of file system B. After applying these changes, file system A then captures a snapshot, snapshot 3 in this example, of the data on that file system. If a previous snapshot of file system A in this example does not exist on file system a, then an entire backup dataset of file system B is generated at file system B, communicated to file system A and restored on file system A" (Paragraph 143).

Regarding claim 98, **Eshel** further teaches a method comprising:
A) copying information concerning rights of users to access data form the source storage device to the target storage device (Paragraphs 54. and 143)

The examiner notes that **Eshel** teaches **"copying information concerning rights of users to access data form the source storage device to the target storage device"** as "Inodes: metadata elements that contain file attributes (e.g., owner, access permissions, modified time, file size), and also

specify the physical disk addresses of data blocks (for small files) or indirect blocks (for large files with more data blocks than the number of disk addresses that fit in an inode). In the description of the exemplary embodiments of present invention, the collection of inodes is referred to as an "node file." The exemplary embodiments store inode files as a regular file (inode plus indirect blocks), but other embodiments use different representations of the collection of inodes. The collection of some or all of the information contained within the inode is referred to as "node information (Paragraph 54) and “After file system A is initialized and becomes the standby file system, file system B then generates, at step 1560, a set of changes between the last snapshot that was received from file system A, snapshot 1 in this example, and communicates that set of changes to file system A. This set of changes contains the snapshot tag pair (A:S1, B:S3). File system A receives, at step 1562, this generated set of changes from file system B and applies those changes to the data stored on file system A in order to establish a copy of the data of file system B. After applying these changes, file system A then captures a snapshot, snapshot 3 in this example, of the data on that file system. If a previous snapshot of file system A in this example does not exist on file system a, then an entire backup dataset of file system B is generated at file system B, communicated to file system A and restored on file system A” (Paragraph 143).

Regarding claim 99, **Eshel** further teaches a system comprising:

A) a second processor configured to: receive, from the host device, at least one data processing request, while the de-migration process is executing (Paragraph 127); and

B) send the at least one data processing request to the target storage device (Paragraphs 127 and 130).

The examiner further that **Eshel** teaches **“a second processor configured to: receive, from the host device, at least one data processing request, while the de-migration process is executing”** as “Another common

use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"send the at least one data processing request to the target storage device"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127) and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130).

Regarding claim 100, **Eshel** further teaches a method comprising:

A) wherein: the request is received from the host side device (Paragraph 127); and

B) the method further comprising: receiving requested data from the copied data (Paragraphs 127 and 130); and

C) providing the requested data to the host device (Paragraphs 127 and 130).

The examiner further that **Eshel** teaches **"wherein: the request is received from the host side device"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"the method further comprising: receiving**

requested data from the copied data" as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127) and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130). The examiner further notes that **Eshel** teaches **"providing the requested data to the host device"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127) and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in

order to bring the standby file system up to the state of the new snapshots captured on the original file system" (Paragraph 130).

8. Claims 22, 26, 55, and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eshel et al.** (U.S. PGPUB 2003/0158862) and in view of **Hubis et al.** (U.S. Patent 6,182,198) as applied to claims 19-21, 23-25, 52-54, 56-58, 79, 88, 90, 92-96, and 98-100 above, and further in view of **Prahlad et al.** (U.S. PGPUB 2006/0010154).

9. Regarding claims 22 and 55, **Eshel** and **Hubis** do not explicitly teach a method and system comprising:

A) wherein the target storage device comprises a NAS filer.

Prahlad, however, teaches **"wherein the target storage device comprises a NAS filer"** as "A NAS device may include a specialized file server or network attached storage system that connects to the network. A NAS device often contains a reduced capacity or minimized operating and file management system (e.g., a microkernel) and normally processes only input/output (I/O) requests by supporting common file sharing protocols such as the Unix network file system (NFS), DOS/Windows, and server message block/common Internet file system (SMB/CIFS). Using traditional local area network protocols such as Ethernet and transmission control protocol/internet protocol (TCP/IP), a NAS device typically enables additional storage to be quickly added by connecting to a network hub or switch" (Paragraph 12) and "The present invention provides, among other things, systems and methods for performing storage operations for electronic data in a computer network on a network attached storage device (NAS). Some of the steps involved in one aspect of the invention may include receiving electronic data from a network device for writing to the NAS device; writing the electronic data to the NAS device in a first location (i.e., primary storage); subsequently storing the electronic data to a second location (i.e., secondary storage); and storing a stub file at the first location, the stub file including a pointer to the second location that may redirect the network device to

the second location if an access request for the electronic data is received from the network device" (Paragraph 17).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Prahlad's** would have allowed **Eshel's** and **Hubis's** to provide a method to intercept file system calls in NAS devices, as noted by **Prahlad** (Paragraph 15).

Regarding claims 26 and 59, **Eshel** and **Hubis** do not explicitly teach a method and system comprising:

A) wherein the selected information in a respective target data file identifies a physical location of the corresponding source data file on the source storage device.

Prahlad, however, teaches **"wherein the selected information in a respective target data file identifies a physical location of the corresponding source data file on the source storage device"** as "A stub file may contain some basic information to identify the file itself and also include information indicating the location of the data on the secondary storage device" (Paragraph 14).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Prahlad's** would have allowed **Eshel's** and **Hubis's** to provide a method to intercept file system calls in NAS devices, as noted by **Prahlad** (Paragraph 15).

10. Claim 89 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Eshel et al.** (U.S. PGPUB 2003/0158862) and in view of **Hubis et al.** (U.S. Patent 6,182,198) as applied to claims 19-21, 23-25, 52-54, 56-58, 79, 88, 90, 92-96, and 98-100 above, and further in view of **George** (U.S. Patent 6,993,679).

11. Regarding claim 89, **Eshel** does not explicitly teach a method comprising:

A) determining an amount of resources that are available to operate the de-migration procedure; and

B) operating the de-migration procedure, only if sufficient resources are available to operate the de-migration procedure.

George, however, teaches **"determining an amount of resources that are available to operate the de-migration procedure"** as "Note that in an alternative embodiment of a method of performing a backup, the backup procedure may progress normally, without first checking for the existence of any entries in the non-read list. If a read error is detected and the read error indicates that the address of the attempted read is on the non-read list, the backup may be paused and the data at that address may be restored (e.g., a system administrator may restore the data from another backup disk). Once the data has been restored and the address has been cleared from the non-read list, the backup may proceed" (Column 7, lines 14-23), and **"operating the de-migration procedure, only if sufficient resources are available to operate the de-migration procedure"** as "Note that in an alternative embodiment of a method of performing a backup, the backup procedure may progress normally, without first checking for the existence of any entries in the non-read list. If a read error is detected and the read error indicates that the address of the attempted read is on the non-read list, the backup may be paused and the data at that address may be restored (e.g., a system administrator may restore the data from another backup disk). Once the data has been restored and the address has been cleared from the non-read list, the backup may proceed" (Column 7, lines 14-23).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **George's** would have allowed **Eshel's** and **Hubis's** to provide a method to prevent requested data from being corrupt in a migration procedure, as noted by **George** (Column 2 , lines 18-22).

12. Claim 97 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Eshel et al.** (U.S. PGPUB 2003/0158862) and in view of **Hubis et al.** (U.S.

Patent 6,182,198) as applied to claims 19-21, 23-25, 52-54, 56-58, 79, 88, 90, 92-96, and 98-100 above, and further in view of **Lam** (U.S. Patent 5,564,037).

13. Regarding claim 97, **Eshel** and **Hubis** do not explicitly teach a method comprising:

A) copying the corresponding source data file from the source storage device to the target storage device, only if the size of the corresponding source data file does not exceed a predetermined limit.

Lam, however, teaches **"copying the corresponding source data file from the source storage device to the target storage device, only if the size of the corresponding source data file does not exceed a predetermined limit"** as "Using the water marks parameter, for example, the HSM system 2 could migrate files from the file server 10 to the secondary storage 20 when the storage space available at the file server 10 reached a critical water mark, at which point emergency migration would immediately occur in accordance with predetermined migration criteria to avoid a "volume full" situation. Files then would be migrated until the storage space available reached a high water mark (e.g., a safe level). The high water mark is defined, for example, as a percentage of the utilized space on the file server 10. When the utilized space is below the critical water mark and above the high water mark, files will be migrated at a predetermined time, for example, on a least recently accessed basis until a low water mark is reached. A low water mark is also defined, for example, as a percentage of the utilized space on the file server 10. When the utilized space is below the low water mark, no migration occurs from the file server 10" (Column 5, lines 54-67-Column 6, lines 1-4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Lam's** would have allowed **Eshel's** and **Hubis's** to better represent the actual properties of the original file, as noted by **Lam** (Column 2 , lines 35-39).

Response to Arguments

14. Applicant's arguments with respect to claims 19-26, 52-59, 79, ands 88-90, have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Article entitled "Data Migration Solution" by **Falconstor** on 23 January 2003. The subject matter disclosed therein is pertinent to that of claims 19-26, 52-59, 79, ands 88-90 (e.g., methods for data migration using stub files with NAS devices).

U.S. Patent 5,564,037 issued to **Lam** on 08 October 1996. The subject matter disclosed therein is pertinent to that of claims 19-26, 52-59, 79, ands 88-90 (e.g., methods for data migration using stub files with NAS devices).

U.S. Patent 5,991,753 issued to **Wilde** on 23 November 1999. The subject matter disclosed therein is pertinent to that of claims 19-26, 52-59, 79, ands 88-90 (e.g., methods for data migration using stub files with NAS devices).

U.S. PGPUB 2005/0015409 issued to **Cheng et al.** on 20 January 2005. The subject matter disclosed therein is pertinent to that of claims 19-26, 52-59, 79, ands 88-90 (e.g., methods for data migration using stub files with NAS devices).

U.S. Patent 7,103,740 issued to **Colgrove et al.** on 05 September 2006. The subject matter disclosed therein is pertinent to that of claims 19-26, 52-59, 79, ands 88-90 (e.g., methods for data migration using stub files with NAS devices).

U.S. PGPUB 2005/0033800 issued to **Kavuri et al.** on 10 February 2005. The subject matter disclosed therein is pertinent to that of claims 19-26, 52-59, 79, ands 88-90 (e.g., methods for data migration using stub files with NAS devices).

U.S. Patent 7,263,590 issued to **Todd et al.** on 10 February 2005. The subject matter disclosed therein is pertinent to that of claim 89 (e.g., methods to pause data backups).

Contact Information

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

December 29, 2008
/Mahesh H Dwivedi/
Examiner, Art Unit 2168

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168